Deep Reinforcement Learning

2025 Second Semester, M.Tech (AIML)

# Session #1: Introduction to the Course

Instructors, Deep Reinforcement Learning Course

Some content for the slides may have been obtained from prescribed books and various other source on the Internet. The authors , hereby acknowledge all the contributors for their material and inputs and gratefully acknowledge those who made their course materials freely available online.

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

# Agenda

- Course Introduction
  Outline of Course, Evaluation & Operation

- Introducing Reinforcement Learning

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

## What is Reinforcement Learning ?

- reward based learning / feedback based learning
- not a type of NN nor it is an alternative to NN. Rather it is an approach for learning
- Autonomous driving, gaming

## Why Reinforcement Learning ?

- a goal-oriented learning based on interaction with environment

# Course Objectives

**Course Objectives**:

1. Understand
    a. the conceptual, mathematical foundations of deep reinforcement learning
    b. various classic & state of the art Deep Reinforcement Learning algorithms
2. Implement and Evaluate the deep reinforcement learning solutions to various problems like planning, control and decision making in various domains
3. Provide conceptual, mathematical and practical exposure on DRL
    a. to understand the recent developments in deep reinforcement learning and

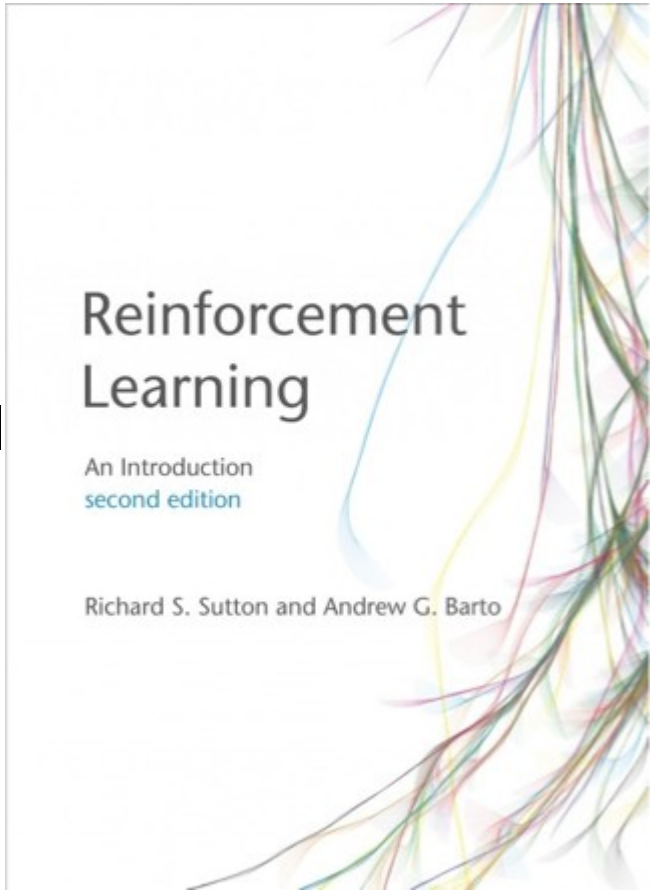
# Learning Outcomes

1. Understand the fundamental concepts of reinforcement learning (RL), algorithms and apply them for solving problems including control, decision-making, and planning.
2. Implement DRL algorithms, handle challenges in training due to stability and convergence
3. Evaluate the performance of DRL algorithms, including metrics such as sample efficiency, robustness and generalization.
4. Understand the challenges and opportunities of applying DRL to real-world problems & model real life problems

# Course Operation

- **Textbooks**

  1. Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, Second Ed. , MI



# Course Operation

- **Evaluation**

  **Two Quizzes for 5% each**; **Best of two will be taken for 5%** ( in final grading);

  Whatever be the points set for quizzes, the score will be scaled to 5%

  NO MAKEUP, for whatever be the reason. Ensure to attend at least one of the quizzes.

  **Two Assignments** - Tensorflow/ Pytorch / OpenAI Gym Toolkit → 25 %

  Assignment 1: Partially Numerical + Implementation of Classic Algorithms - 10%

  Assignment 2: Deep Learning based RL - 15%

  Mid-Term Exam - 30% [ Only to be written in A4 pages, scanned and uploaded]

  Comprehensive Exam - 40% [ Only to be written in A4 pages, scanned and uploaded]

- **Webinars/Tutorials**

  4 tutorials : 2 before mid-sem & 2 after mid-sem

# Course Operation

- Schedule of Schedule of Quizzes

  See the announcements for details

- Schedule of Assignments

  See the announcements for details

- Schedule of Webinars

  See the announcements for details

# Course Operation

- How to reach us ? (for any question on lab aspects, availability of slides on portal, quiz availability , assignment operations )

  See the announcements for details

- **Plagiarism [ Important ]**

All submissions for graded components must be the result of your original effort. It is strictly prohibited to copy and paste verbatim from any sources, whether online or from your peers. The use of unauthorized sources or materials, as well as collusion or unauthorized collaboration to gain an unfair advantage, is also strictly prohibited. Please note that we will not distinguish between the person sharing their resources and the one receiving them for plagiarism, and the consequences will apply to both parties equally.
In cases where suspicious circumstances arise, such as identical verbatim answers or a significant overlap of unreasonable similarities in a set of submissions, will be investigated, and severe punishments will be imposed on all those found guilty of plagiarism.

# Reinforcement Learning

***Reinforcement learning (RL) is based on rewarding desired behaviors or punishing undesired ones. Instead of one input producing one output, the algorithm produces a variety of outputs and is trained to select the right one based on certain variables – Gartner***

## When to use RL?

RL can be used in large environments in the following situations:

1. A model of the environment is known, but an analytic solution is not available;
2. Only a simulation model of the environment is given (the subject of simulation-based optimization)
3. The only way to collect information about the environment is to interact with it.

# (Deep) Reinforcement Learning

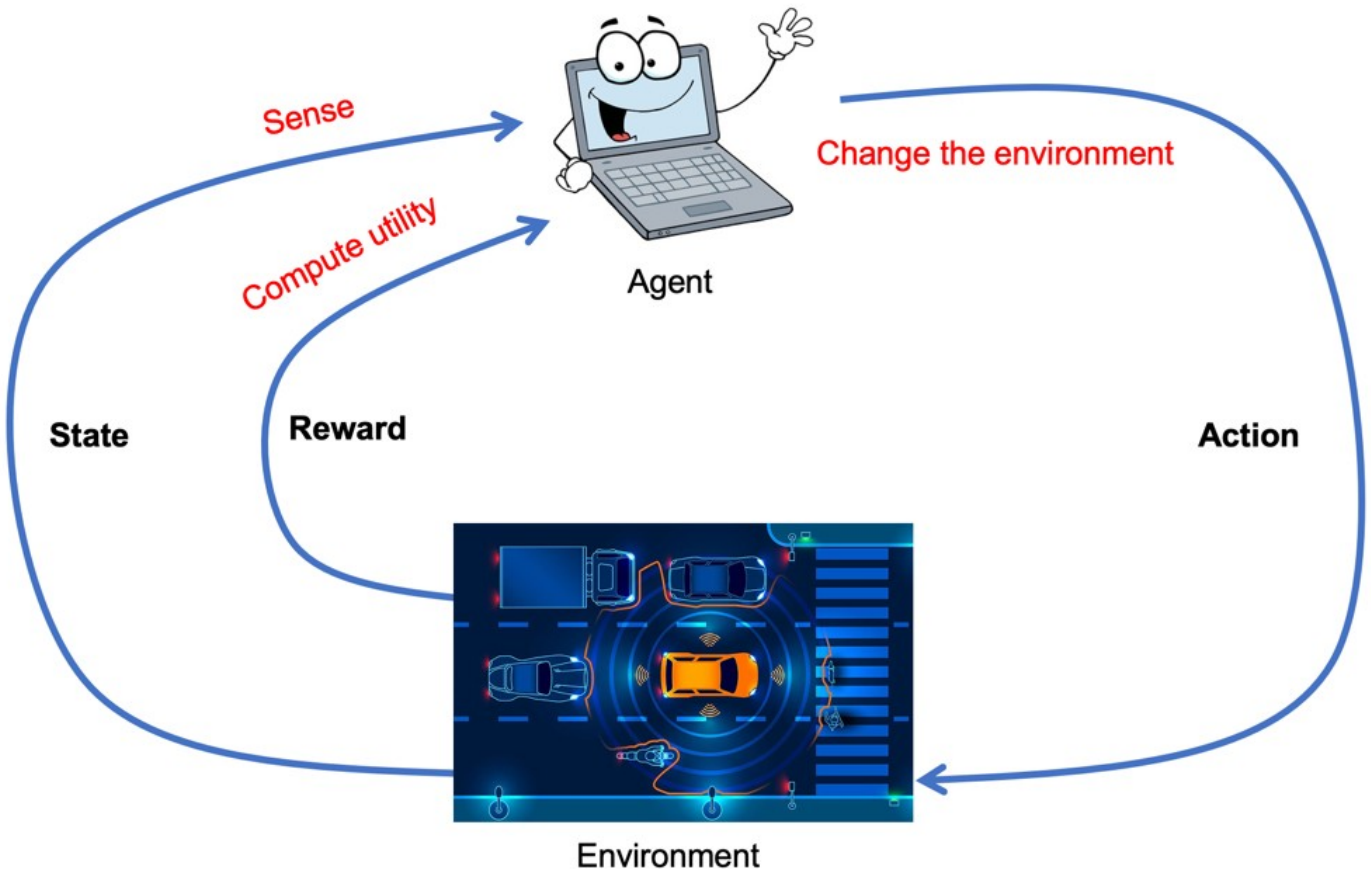
| | | | |
|---|---|---|---|
| Paradigm | Supervised Learning | Unsupervised Learning | Reinforcement Learning |
| Objective | $p_\theta(y\|x)$ | $p_\theta(x)$ | $\pi_\theta(a\|s)$ |
| Applications | → Classification<br>→ Regression | → Inference<br>→ Generation | → Prediction<br>→ Control |

# Types of Learning

| Criteria | Supervised ML | Unsupervised ML | Reinforcement ML |
|---|---|---|---|
| *Definition* | Learns by using labelled data | Trained using unlabelled data without any guidance. | Works on interacting with the environment |
| *Type of data* | Labelled data | Unlabelled data | No – predefined data |
| *Type of problems* | Regression and classification | Association and Clustering | Exploitation or Exploration |
| *Supervision* | Extra supervision | No supervision | No supervision |
| *Algorithms* | Linear Regression, Logistic Regression, SVM, KNN etc. | K – Means, C – Means, Apriori | Q – Learning, SARSA |
| *Aim* | Calculate outcomes | Discover underlying patterns | Learn a series of action |

# Reinforcement learning

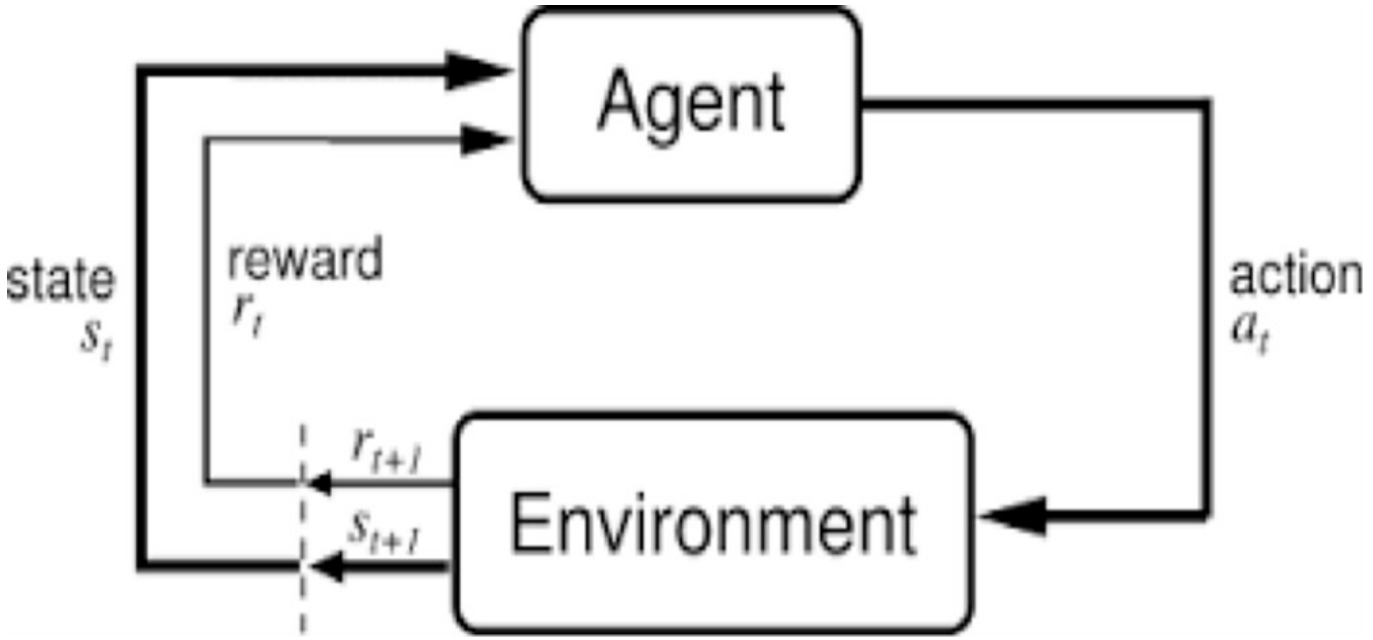
- Basic idea:
  - Receive feedback in the form of rewards
  - Agent's utility is defined by the sum of rewards
  - Should (learn to) act in a way that maximizes expected rewards
  - Learning is based on observed action outcomes



# Characteristics of RL

- No supervision, only a real value or reward signal
- Decision making is sequential
- Time plays a major role in reinforcement problems
- Feedback isn’t prompt but delayed

# Elements of Reinforcement Learning




**Beyond the agent and the environment, one can identify four main sub-elements of a reinforcement learning system: *a policy*, a *reward* , a *value* function, and, optionally, a *model* of the environment.**

# Elements of Reinforcement Learning

- **Agent**
  - An **entity** that tries to learn the best way to perform a specific task.
  - In our example, the child is the agent who learns to ride a bicycle.

- **Action (A)** -
  - **What the agent does** at each time step.
  - In the example of a child learning to walk, the action would be “walking”.
  - A is the set of all possible moves.
  - In video games, the list might include running right or left, jumping high or low, crouching or standing still.

# Elements of Reinforcement Learning

- **State (S)**
  - **Current situation** of the agent.
  - After doing performing an action, the agent can move to different states.
  - In the example of a child learning to walk, the child can take the action of taking a step and move to the next state (position).
- **Rewards (R)**
  - Feedback that is given to the agent based on the action of the agent.
  - If the action of the agent is good and can lead to winning or a positive side then a positive reward is given and vice versa.

# Elements of Reinforcement Learning

- **Environment**
  - Outside world of an agent or physical world in which the agent operates.

Formal Definition - ***Reinforcement learning*** *(**RL**) is an area of machine learning concerned with how intelligent **agents** ought to take **actions** in an **environment** in order to maximize the notion of cumulative **reward**.*

# Elements of Reinforcement Learning

***Goal of RL*** - maximize the total amount of rewards or cumulative rewards that are received by taking actions in given states.

- Notations –

  a set of states as $S$,

  a set of actions as $A$,

  a set of rewards as $R$.

At each time step $t = 0, 1, 2, \ldots$, some representation of the environment's state $S_t \in S$ is received by the agent. According to this state, the agent selects an action $A_t \in A$ which gives us the state-action pair $(S_t, A_t)$. In the next time step $t+1$, the transition of the environment happens and the new state $S_{t+1} \in S$ is achieved. At this time step $t+1$, a reward $R_{t+1} \in R$ is received by

# Elements of Reinforcement Learning

## Policy (π)

- Policy in RL decides which action will the agent take in the current state.
- It tells the *probability that an agent will select a specific action from a specific state*.
- Policy is a function that maps a given state to probabilities of selecting each possible action from the given state.

•If at time ***t,*** an agent follows policy ***π***, then ***π(a|s)*** becomes the probability that the action at time step ***t*** is $a_{t=a}$ if the state at time step ***t*** is $S_{t=s}$ .The meaning of this is, the probability that an agent will take an action ***a*** in state ***s*** is ***π(a|s)*** at time ***t*** with policy **π**.

# Elements of Reinforcement Learning

**Value Functions**

- a simple measure of how good it is for an agent to be in a given state, or how good it is for the agent to perform a given action in a given state.

Two types

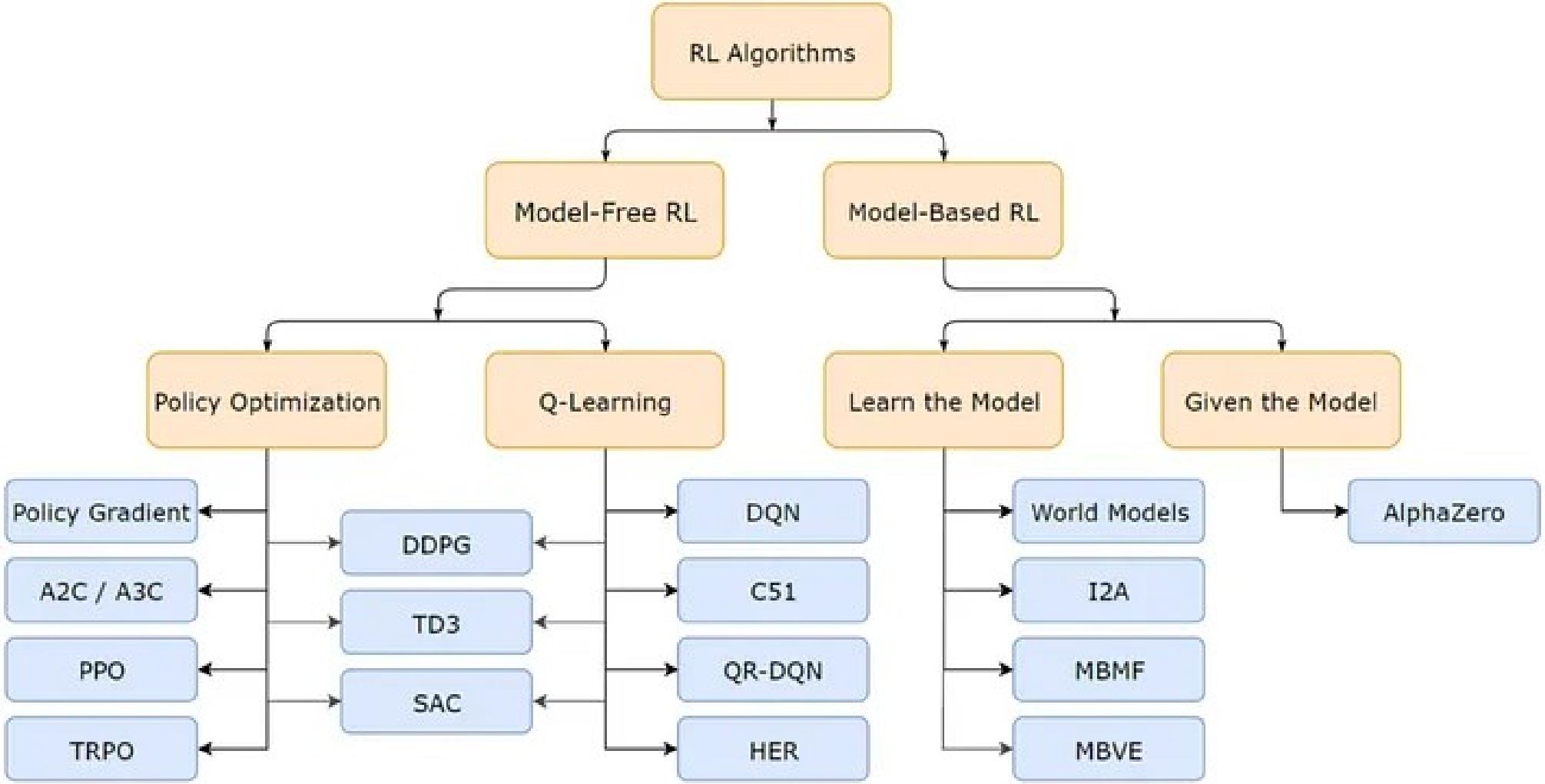
- State- value function
- Action-value function

# Elements of Reinforcement Learning

**Model of the environment**

- mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave.
- For example, given a state and action, the model might predict the resultant next state and next reward.
- Models are used for planning

# (Deep) Reinforcement Learning




From OpenAI

# Tic-Tac-Toc

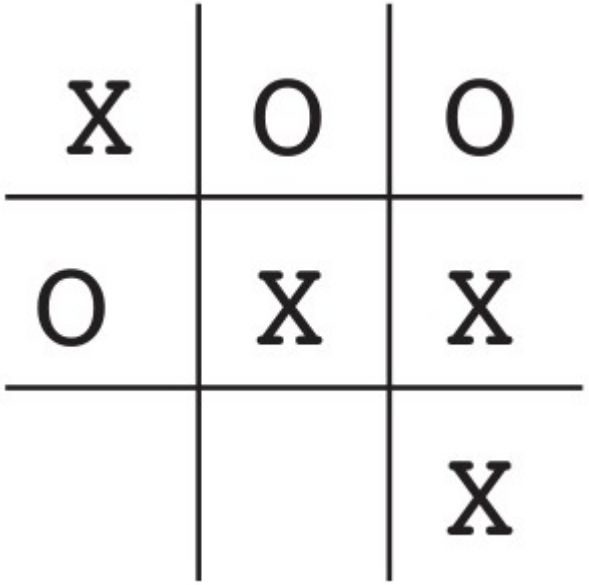
Two players take turns playing on a three-by-three board. One player plays Xs and the other Os until one player wins by placing three marks in a row, horizontally, vertically, or diagonally

**Assumptions**

- playing against an imperfect player, one whose play is sometimes incorrect and allows you to win

**Aim**

*How might we construct a player that will find the imperfections in its opponent's play and learn to maximize its chances of winning?*



# Tic-Tac-Toc

| States | Initial Values |
|---|---|
| $\begin{bmatrix} X & & \\ & & \\ & & \end{bmatrix}$ | 0.5 |
| $\begin{bmatrix} X & O & O \\ & X & \\ & & \end{bmatrix}$ | 0.5 |
| $\begin{bmatrix} X & O & O \\ & X & \\ & & X \end{bmatrix}$ | 1.0 |
| $\begin{bmatrix} X & & O \\ X & & O \\ & X & O \end{bmatrix}$ | 0 |
| ... | ... |

**Learning Task:** Play as many times against the opponent and learn the values

| | | |
|---|---|---|
| X | O | O |
| O | X | X |
| | | X |

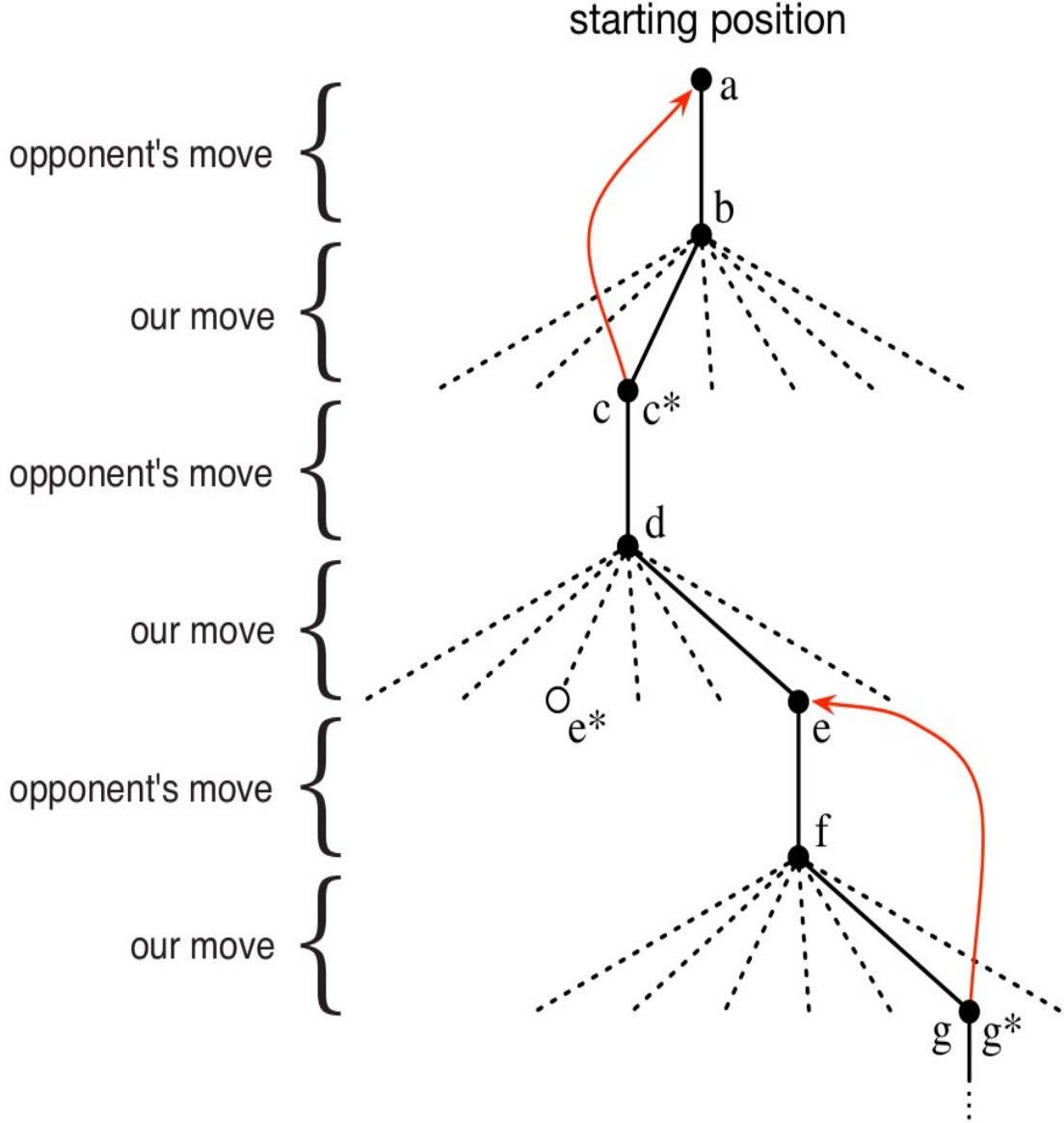
Set up a table of states initial values

# Tic-Tac-Toc

| States | Initial Values |
|---|---|
| $\begin{bmatrix} X & & \\ & & \\ & & \end{bmatrix}$ | 0.5 |
| $\begin{bmatrix} X & O & O \\ & X & \\ & & \end{bmatrix}$ | 0.5 |
| $\begin{bmatrix} X & O & O \\ & X & \\ & & X \end{bmatrix}$ | 1.0 |
| $\begin{bmatrix} X & & O \\ X & & O \\ & X & O \end{bmatrix}$ | 0 |
| ... | ... |

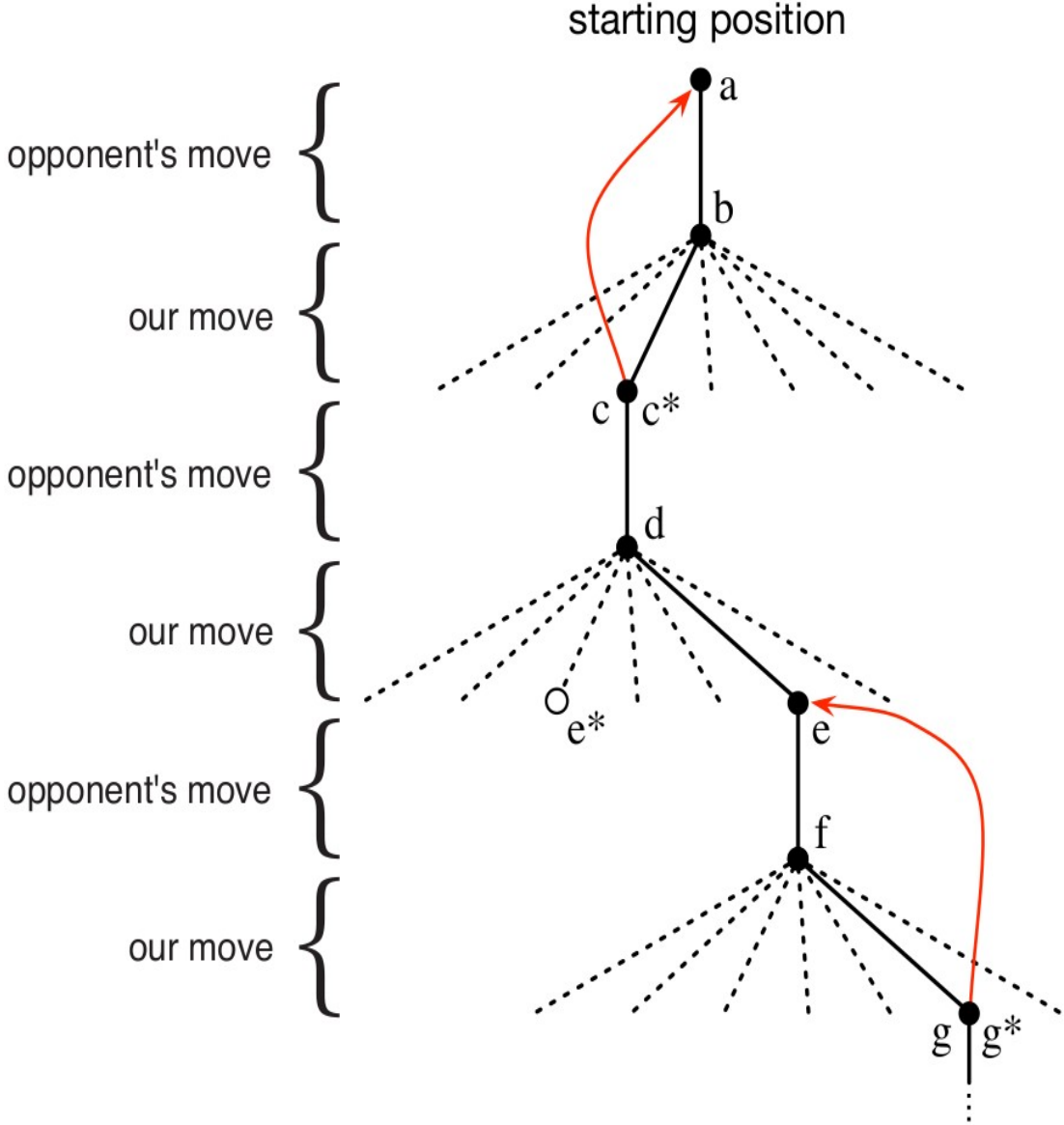
$S_t$ - state before greedy move

$S_{t+1}$ - state after greedy move




$$V(S_t) \leftarrow V(S_t) + \alpha\left[V(S_{t+1}) - V(S_t)\right]$$

# Tic-Tac-Toc




**Temporal Difference Learning Rule**

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[V(S_{t+1}) - V(S_t)\Big]$$

**- Step Size Parameter**

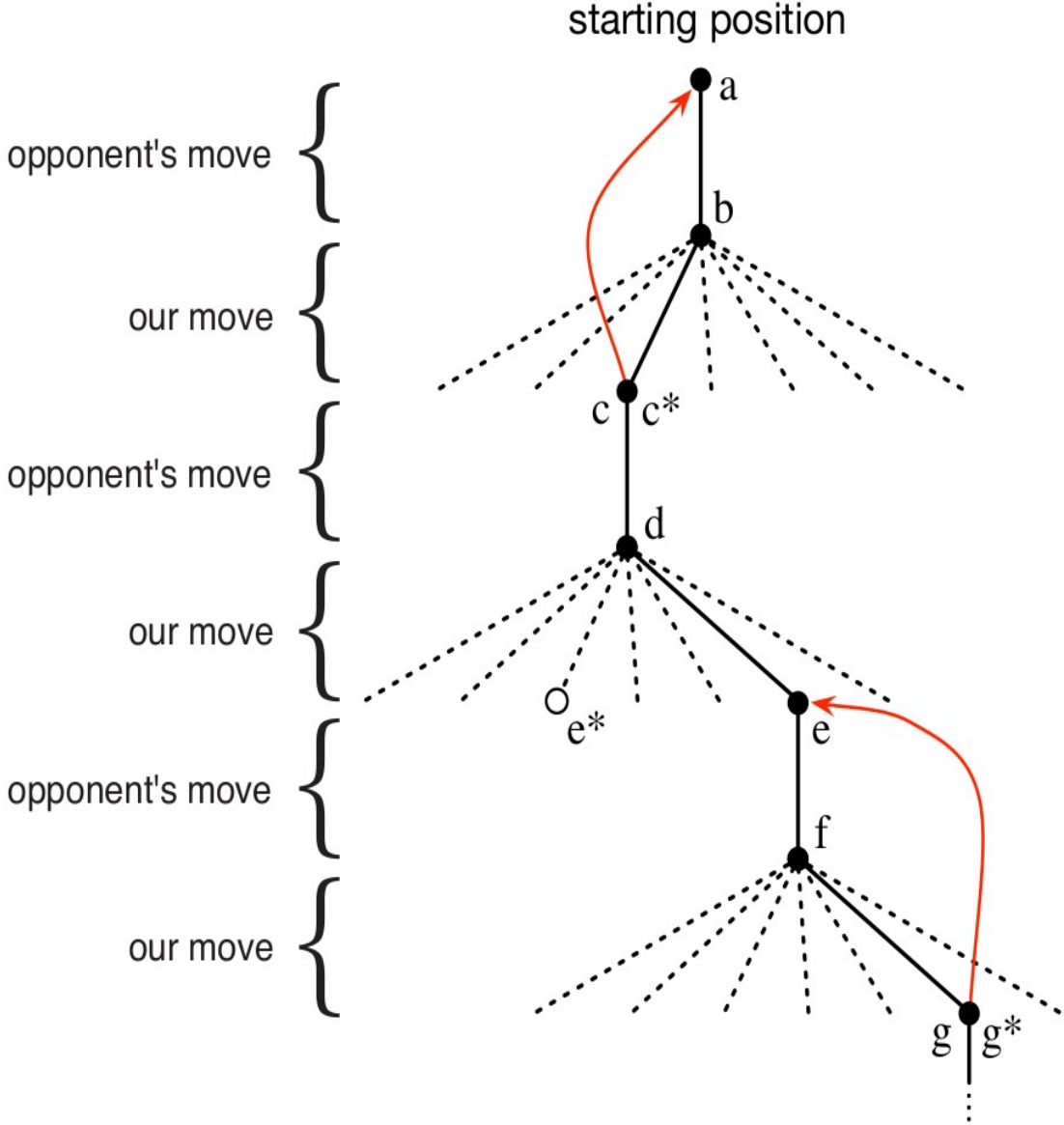
# Tic-Tac-Toc

Questions:

(1) What happens if □ is gradually made to 0 over many games with the opponent?

(2) What happens if □ is gradually reduced over many games, but never made 0?

(3) What happens if □ is kept constant throughout its life time?




**Temporal Difference Learning Rule**

$$V(S_t) \leftarrow V(S_t) + \alpha\left[V(S_{t+1}) - V(S_t)\right]$$

**□- Step Size Parameter**

# Tic-Tac-Toc

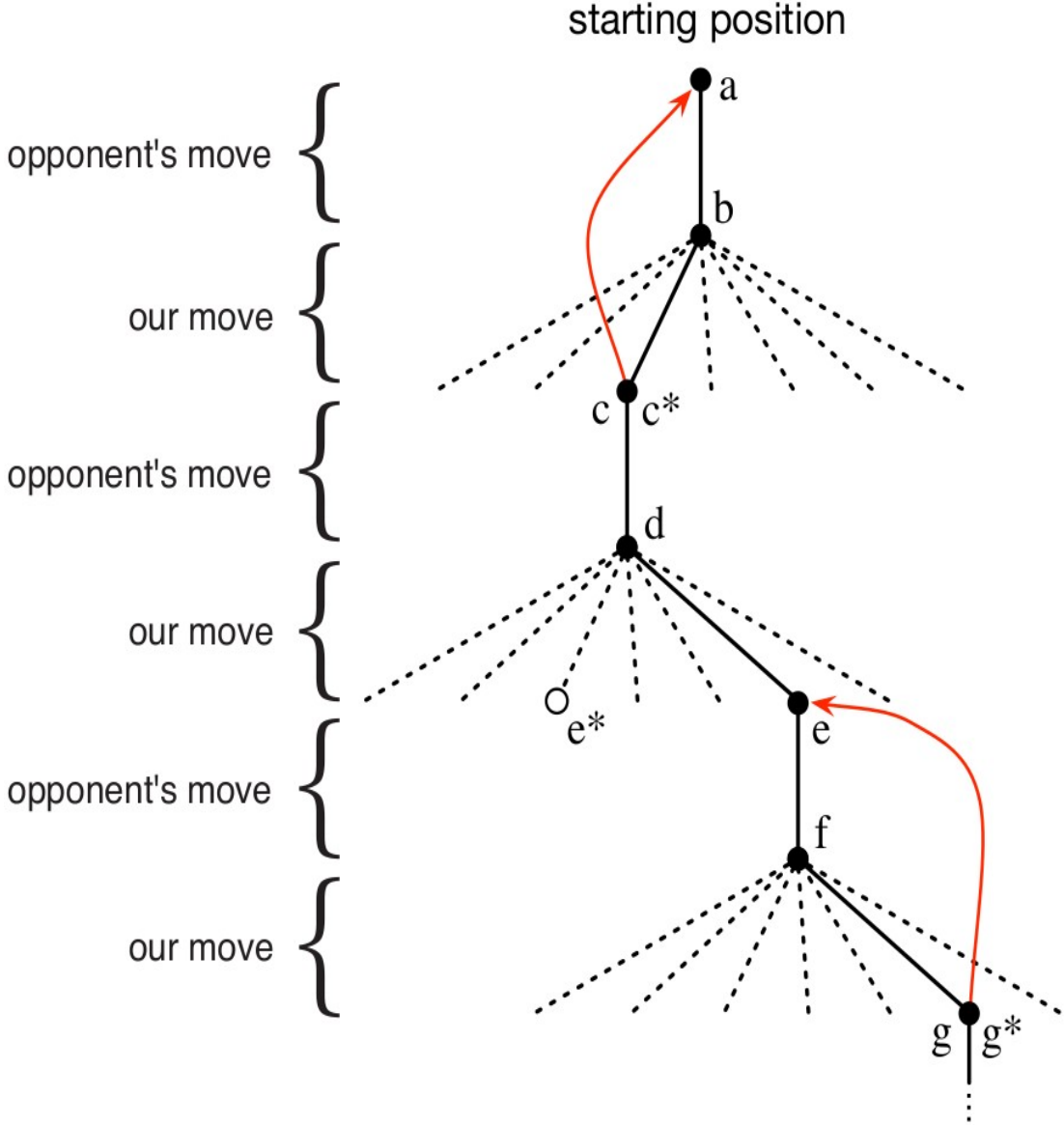
**Key Takeaways:**

(1)Learning while interacting with the environment (opponent).

(2)We have a clear goal

(3)Our policy is to make moves that maximizes our chances of reaching goal

- Use the values of states most of the time (exploration) and explore rest of the time.



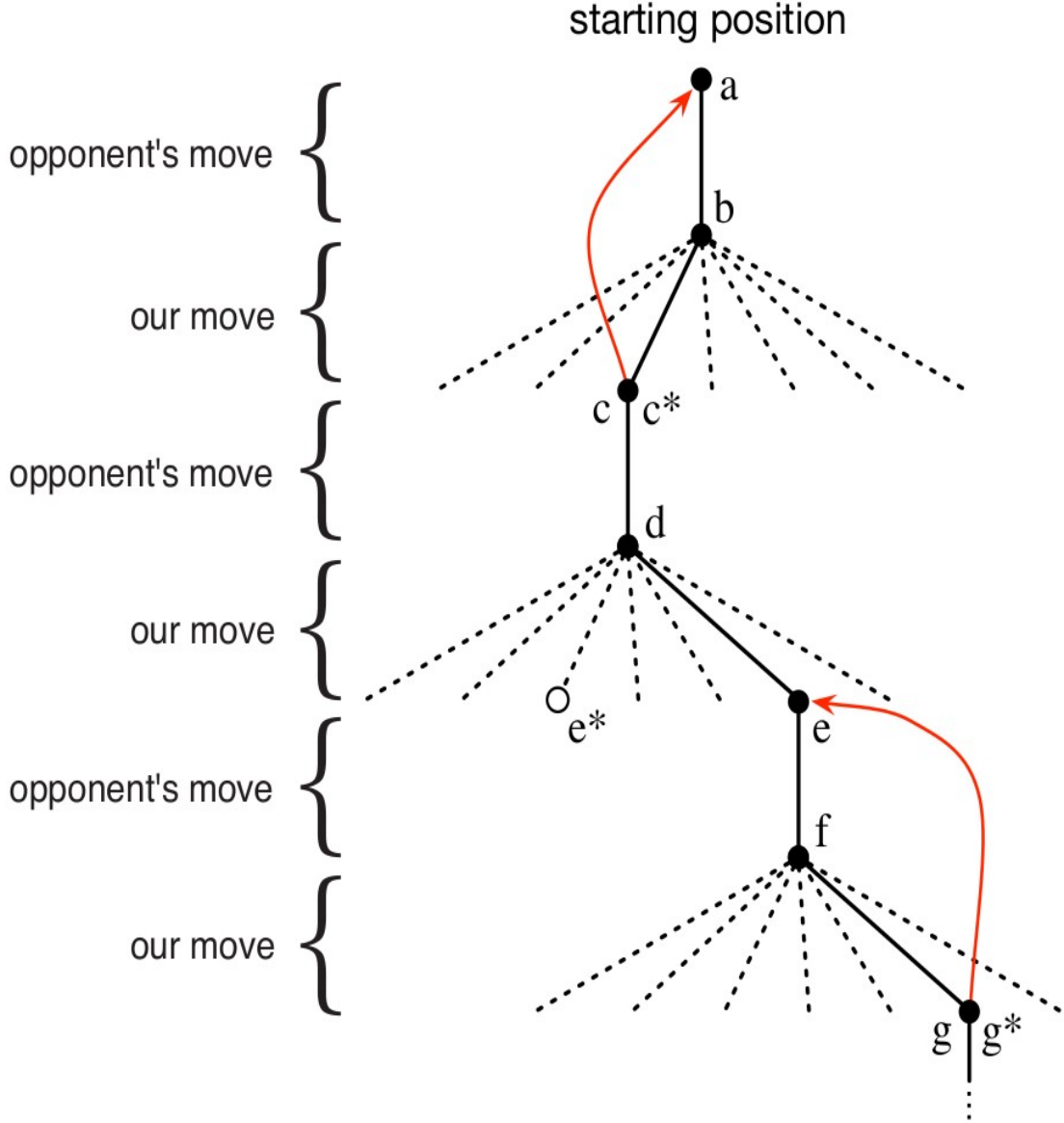

**Temporal Difference Learning Rule**

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[V(S_{t+1}) - V(S_t)\Big]$$

**- Step Size Parameter**

# Tic-Tac-Toc

**Reading Assigned:**
Identify how this reinforcement learning solution is different from solutions using minimax algorithm and genetic algorithms.
Post your answers in the discussion forum;



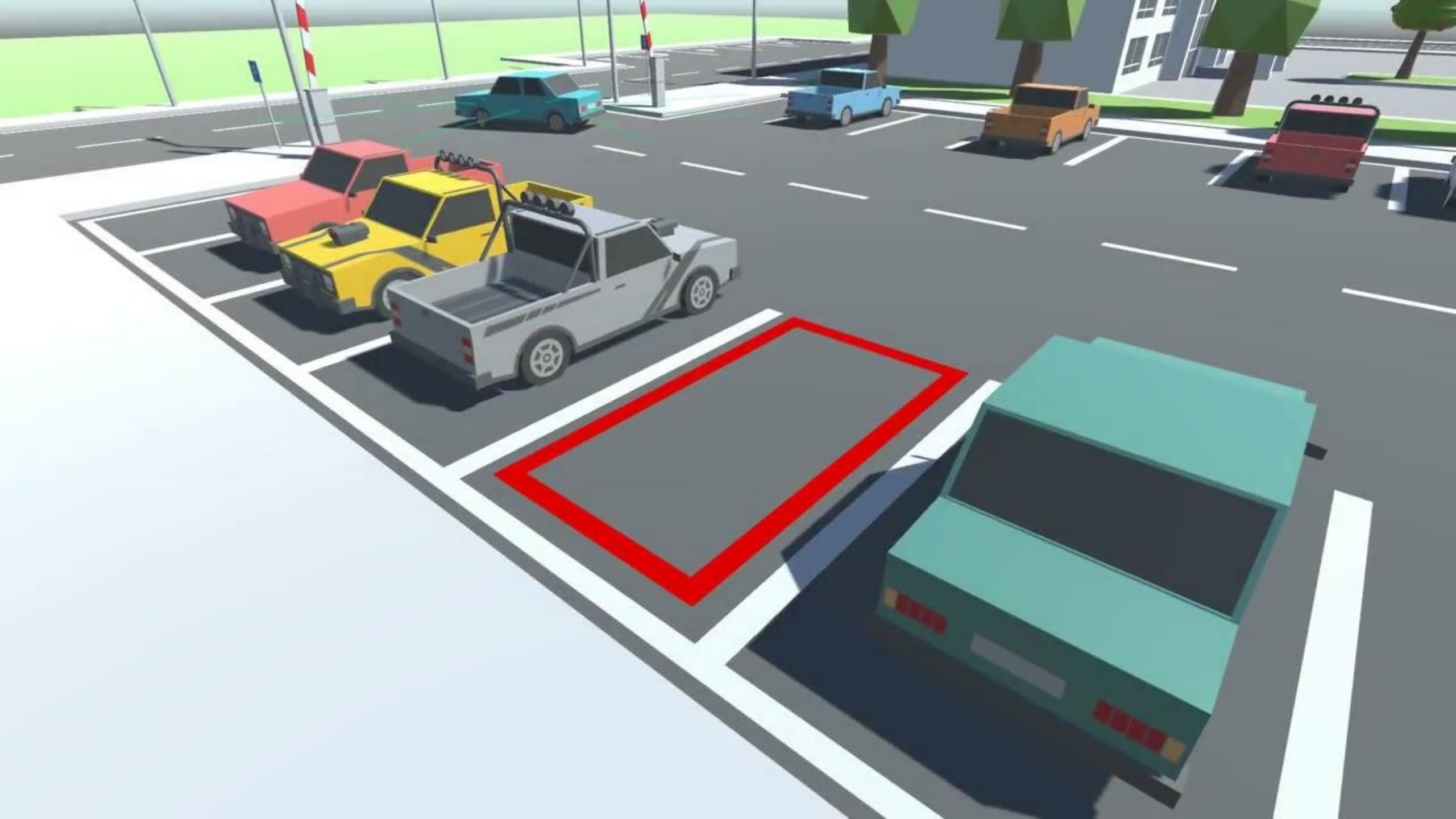

**Temporal Difference Learning Rule**

$$V(S_t) \leftarrow V(S_t) + \alpha\left[V(S_{t+1}) - V(S_t)\right]$$

**- Step Size Parameter**

# Autonomous driving

- **Objective**: reach destination safely
- **State**: image from front camera
- **Action**: Steering, acceleration, brake
- **Reward**: forward speed
  - Drive off road = -10
  - Hit a person = -1000
  - Drive at speed limit = +2
  - Align with lane = +4
  - Run a stop sign = -5
  - Erratic driving= -3
  - …



# References for today’s session

1. Chapter 1 - Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, Second Ed. , MIT Press

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

# End of Session #1

# Thank you